ECE 595Z
Digital Systems Design Automation
Module 9 (Lectures 29-30): Low Power Design
Lecture 30

Anand Raghunathan
MSEE 348
raghunathan@purdue.edu
Lecture 29: Re-cap

• Virtually every IC design today is impacted by power consumption

• Role of design automation
  – Power estimation
  – Power reduction

• Power estimation at the logic level
  – Power depends on values and transitions at signals in the circuit
Approaches to Power Estimation

• **Simulation-based**
  – Given (user-provided) test benches
    • Simulate the circuit (modeling gate delays)
      – Online: Evaluate power models during simulation
      – Offline: Record switching activities and signal probabilities during simulation and post-process to estimate power

• **Problem:** Long test benches may be necessary, can be very slow

• **Solutions**
  – Statistical sampling of input traces
    • Still requires simulation of shorter traces
  – **Probabilistic** analysis of circuits
    • Static (does not require simulation)
Approaches to Power Estimation

- **Probabilistic analysis**
  - Compute “average” switching activities and value probabilities at internal signals in the circuit
  - Evaluate power models

- **Advantage**: No simulation needed (fast), no need for test bench

\[ N_{z_i} = \frac{\sum_{\text{all } v_1, v_2} N_{z_i}(v_1, v_2)}{2^{\#\text{inputs}} \times 2^{\#\text{inputs}}} \]
Value Probabilities

• Define $P_x$ as the probability that signal $x=1$
• Value probability at a gate output can be computed using probabilities at its inputs

**AND gate:** $P_I = P_A \cdot P_B$
**OR gate:** $P_J = P_B + P_C$

**Assumption:**
A, B, C are uncorrelated (we will re-visit this since it is not valid for sequential circuits)
Value Probabilities: Spatial Correlation

- In general, the inputs to a gate may be correlated
  - Failure to account for the correlation will lead to an incorrect estimate for the probability at the gate output

I and J are correlated since both depend on B

\[ P_K \neq P_I \times P_J \]
Computing Value Probabilities Considering Spatial Correlation

- Procedure for computing value probability considering correlations
  1. Write a Boolean expression for the signal in terms of primary inputs
  2. Convert the expression into a disjoint SOP expression (cubes are pair-wise disjoint, i.e., intersection is NULL)
  3. Compute the output probability for the disjoint SOP expression

\[ F = ab + bc \]
\[ F = ab + a'bc \]
\[ P_F = P_a P_b + P_b P_c \]
\[ P_F = P_a P_b + (1 - P_a) P_b P_c \]

Disjoint cubes can be obtained from a representation
Computing Switching Activities Considering Gate Delays and Correlations

- First, consider zero-delay model
  - All gate outputs switch instantaneously after input vector is applied

\[
S_f(t=T) = f(t=0) \oplus f(t=T) = (a(0) b(0)) \oplus (a(T) b(T))
\]

\[
S_h(t=T) = h(t=0) \oplus h(t=T) = (a(0) b(0) + c(0)) \oplus (a(T) b(T) + c(T))
\]

Under the zero-delay model, switching activity at a gate output is just the probability of the switching function evaluating to True
Computing Switching Activities Considering Gate Delays and Correlations

Now, consider general delay model

\[ f(0) = a(0) b(0) \]
\[ f(T) = a(0) b(0) \]
\[ f(T+1) = a(T) b(T) \]

\[ h(0) = f(0) + c(0) \]
\[ h(T) = f(0) + c(0) \]
\[ h(T+1) = f(T) + c(T) \]
\[ h(T+2) = f(T+1) + c(T) \]

Switching functions

\[ S_{f}(1) = f(T) \oplus f(T+1) \]
\[ N_{f} = P(S_{f}(1)) \]

\[ S_{h}(1) = h(T) \oplus h(T+1) \]
\[ S_{h}(2) = h(T+1) \oplus h(T+2) \]

\[ N_{h} = P(S_{h}(1)) + P(S_{h}(2)) \]

Switching activity for a signal is the sum of all its switching function probabilities
Computing Switching Activities Considering Gate Delays and Correlations

• For each gate
  – Enumerate potential times at which gate can switch.
  – For each potential switching time
    • Construct gate switching function
      – Boolean expression that represents conditions for gate switching
      – Function of current and previous input vectors
    • Compute probability of the gate switching function evaluating to True
  – Compute switching activity as sum of all switching function probabilities
Sequential Circuits

• Two additional challenges
  – Need to compute probabilities at present state lines
  – Account for temporal correlation between values appearing at PS lines in one cycle and the next

• $PS(T) = f(PS(0), PI(0))$
Computing Present State Input Probabilities

• **Exact method**
  
  – Extract the state transition graph (STG) from the circuit
  – Compute state probabilities by solving Chapman-Kolmogorov equations

\[
\text{prob}(s_i) = \sum_{m} \text{prob}(s_m) \times \text{prob}(\text{edge}_{m \rightarrow i})
\]

\[
\sum_{i=1}^{K} \text{prob}(s_i) = 1
\]

– Use state probabilities and state encoding to compute present state input probabilities
Computing Present State Input Probabilities: Example

\[
\begin{align*}
\text{prob}(R) &= 0.5 \times \text{prob}(A) \\
\text{prob}(A) &= 0.5 \times (\text{prob}(R) + \text{prob}(B) + \text{prob}(C)) \\
\text{prob}(B) &= 0.5 \times (\text{prob}(R) + \text{prob}(A)) \\
\text{prob}(R) + \text{prob}(A) + \text{prob}(B) + \text{prob}(C) &= 1
\end{align*}
\]

\[
\begin{align*}
\text{prob}(R) &= 0.167 \quad \text{prob}(A) = 0.333 \\
\text{prob}(B) &= 0.25 \quad \text{prob}(C) = 0.25
\end{align*}
\]
Computing Present State Input Probabilities: Example

\[
\begin{align*}
\text{prob}(R) &= 0.167 \quad \text{prob}(A) = 0.333 \\
\text{prob}(B) &= 0.25 \quad \text{prob}(C) = 0.25 \\
\text{prob}(00) &= 0.167 \quad \text{prob}(01) = 0.333 \\
\text{prob}(10) &= 0.25 \quad \text{prob}(11) = 0.25 \\
\end{align*}
\]

\[
\begin{align*}
\text{prob}(p_{s_1} = 0) &= \text{prob}(00) + \text{prob}(01) = 0.5 \\
\text{prob}(p_{s_1} = 1) &= \text{prob}(10) + \text{prob}(11) = 0.5 \\
\text{prob}(p_{s_2} = 0) &= \text{prob}(00) + \text{prob}(10) = 0.417 \\
\text{prob}(p_{s_2} = 1) &= \text{prob}(01) + \text{prob}(11) = 0.583 \\
\end{align*}
\]

Note that PS lines are correlated. For example:

\[
\text{prob}(p_{s_1} = 0) \times \text{prob}(p_{s_2} = 0) = 0.208 \neq \text{prob}(00)
\]
Computing Present State Input Probabilities

- STG is very large for most practical circuits

  - **Approximate method:** Compute probabilities at PS inputs independent of each other

\[
\begin{align*}
ns_1 &= f_1(i_1, ..., i_M, ps_1, ..., ps_N) \\
n_2 &= f_2(i_1, ..., i_M, ps_1, ..., ps_N) \\
& \vdots \\
n_N &= f_N(i_1, ..., i_M, ps_1, ..., ps_N) \\
\end{align*}
\]

\[
\begin{align*}
\text{prob}(ns_1) &= \text{prob}(f_1(i_1, ..., i_M, ps_1, ..., ps_N)) \\
\text{prob}(ns_2) &= \text{prob}(f_2(i_1, ..., i_M, ps_1, ..., ps_N)) \\
& \vdots \\
\text{prob}(ns_N) &= \text{prob}(f_N(i_1, ..., i_M, ps_1, ..., ps_N))
\end{align*}
\]
Computing Present State Input Probabilities

• Set of non-linear equations
  – Use iterative techniques (Newton-Raphson)

\[
\begin{align*}
\text{prob}(\text{ns}_1) &= \text{prob}(f_1(i_1, \ldots, i_M, ps_1, \ldots, ps_N)) \\
\text{prob}(\text{ns}_2) &= \text{prob}(f_2(i_1, \ldots, i_M, ps_1, \ldots, ps_N)) \\
&\quad \ldots \\
\text{prob}(\text{ns}_N) &= \text{prob}(f_N(i_1, \ldots, i_M, ps_1, \ldots, ps_N))
\end{align*}
\]

In steady-state:

\[
\text{prob}(ps_i = 1) = \text{prob}(\text{ns}_i = 1) = p_i \quad 1 \leq i \leq N
\]
Computing Present State Input Probabilities

Assume probability of input $i$ is 0.5

$$ns_1 = \overline{i} \overline{ps_1} \overline{ps_2} + i \overline{ps_1} ps_2 + \overline{i} ps_1 ps_2 + i ps_1 \overline{ps_2}$$

$$ns_2 = i \overline{ps_1} \overline{ps_2} + ps_1$$

Express NS probabilities in terms of PS probabilities

$$p_1 = 0.5 \left( (1 - p_1) (1 - p_2) + (1 - p_1) p_2 + p_1 p_2 + p_1 (1 - p_2) \right)$$

$$p_2 = 0.5 \left( (1 - p_1) (1 - p_2) + p_1 \right)$$

Solve

$$p_1 = 0.5 \quad p_1 = \text{prob}(ps_1 = 1)$$

$$p_2 = 0.6 \quad p_2 = \text{prob}(ps_2 = 1)$$
Modeling Temporal Correlation at Present State Inputs

- Create an additional copy of the next-state logic to feed the equations that represent the switching functions.
Summary: Power Estimation

• Power estimation requires accurate computation of switching activities and value probabilities at internal signals
  – Simulation is the most commonly used approach, but often too slow
  – Probabilistic analysis leads to one-shot computation of all switching activities and value probabilities
    • Challenges: Spatial correlations due to re-convergent fanout within circuit, computing probabilities at present state inputs, temporal correlation at present state inputs
      – Alternative: Statistical sampling
  
• Both probabilistic and statistical techniques work well for large circuits, and are used in practice
Automatic Power Reduction
All of the above techniques reduce the switched capacitance in the circuit, and are hence complementary to frequency and supply voltage scaling.
Technology Mapping for Low Power

- Key idea: A desirable mapping results in high switching activity nets having low capacitance
  - Get “hidden” inside cells
  - Driven/loaded by smaller gates

<table>
<thead>
<tr>
<th>Cell</th>
<th>Area</th>
<th>O/p Cap.</th>
<th>I/p Cap.</th>
</tr>
</thead>
<tbody>
<tr>
<td>INV</td>
<td>928</td>
<td>0.1029</td>
<td>0.0514</td>
</tr>
<tr>
<td>NAND2</td>
<td>1392</td>
<td>0.1421</td>
<td>0.0747</td>
</tr>
<tr>
<td>AOI22</td>
<td>2320</td>
<td>0.3410</td>
<td>0.1033</td>
</tr>
</tbody>
</table>

Area =
Power =
Technology Mapping for Low Power

• Simple extension of mapping for area
  – Before mapping, compute switching activity at all signals in the subject graph
  – During mapping, use $\text{Power(match)} = \sum_{i \in \text{pins(match)}} C_i N_i$

```c
int min_power_map(v, P){
    /* v is a vertex in the tree, P is the set of pattern graphs */
    best_cost = infinity;
    foreach(m = match(v, P)) {
        cost(m) = Power(m) + \sum_{v_i \in \text{inputs}(m)} \text{min_power_map}(v_i, P);
        if(cost(m) < best_cost){
            match(v) = m;
            best_cost = cost(m);
        }
    }
    return best_cost;
}
```
Technology Mapping for Low Power

• Extensions
  – If $C_i$ (power coefficients) depend on output load, use similar approach as technology mapping for delay
    • Discretize range of possible loads and compute best match for each
  – Considering leakage power:
    • Modify Power(m) to consider leakage in addition to switched capacitance
  – Multi-Vdd and multi-Vth mapping
    • Model as richer library
Clock Gating

- Clock (distribution network + FF loads) can account for a significant portion of the total circuit’s power
  - > 33% in a high-performance microprocessor
  - Huge capacitance (need to route all over the chip, buffers), high activity (2 transitions per cycle)

Power breakdown for LEON2 embedded processor core:

- Total core power (mW), 63, 56%
- Calculated clk pwr, 48.507, 44%
Clock Gating

• Basic idea: Suppress the clock signal from propagating through a part of the clock network
  – Question: When can you do this?
Clock Gating

- Two different approaches
  - $PS_i \oplus NS_i$ tells you when the output of the FF will change
  - Sequential ODCs for the output of a FF tell you when it’s value is not useful
- Approximate (fast) method
  - Look for self-loops and derive sensitization conditions
Pre-computation

• Take clock gating to the next level
• Selectively pre-compute the output of the circuit (using simpler circuits) one cycle in advance, and disable the original circuit

Predictor functions:

\[ g_1 = 1 \Rightarrow f = 1 \]
\[ g_2 = 1 \Rightarrow f = 0 \]
Example: Comparator

- MSBs alone can determine the output if they are unequal.

\[ g_1 = C_{n-1} \overline{D_{n-1}} \]
\[ g_2 = C_{n-1} \overline{D_{n-1}} \]
**Pre-computation: Automatic generation of Conditions**

- **Problem**: Given $f(x_1, \ldots, x_n)$ and $k$, the number of inputs to the pre-computation logic, determine:
  - The inputs $S$ to the precomputation logic
  - The precomputation logic functions $g_1$ and $g_2$
  - Find $S = \{x[1], \ldots, x[k]\}$ that maximizes $\text{prob}(g_1(x[1], \ldots, x[k]) + g_2(x[1], \ldots, x[k]) = 1)$

- Recall the Universal Quantification (consensus) of a function $f$ with respect to a variable $x$
  - $\forall_x(f) = f_x \cap f'_x$

- **Predictor functions can be generated as**
  - $g_1 = \forall_{x[k+1]} x_{[k+2]} \ldots x[n](f)$
  - $g_2 = \forall_{x[k+1]} x_{[k+2]} \ldots x[n](f')$
Summary: Power Reduction

- Logic-level power reduction techniques focus on reducing the switched capacitance
  - Technology mapping for power
  - Automatic generation of clock gating, pre-computation logic, operand isolation logic
  - Commercial state-of-the-art: Technology mapping, automatic clock gating
ECE 595Z
Digital Logic Synthesis:
Electronic Design Automation at the Logic Level
Selected Recent Developments / New Challenges

Anand Raghunathan
MSEE 318
raghunathan@purdue.edu
Outline

• Logic synthesis is a fairly mature area
• New challenges arise due to
  – Technology scaling trends
    • Increase in interconnect delay/power
      – Need for tighter integration with physical design, more accurate models
    • Process variations
  – Increase in circuit complexity
    • Saturation in clock frequencies of computing platforms that run tools
    • Need to parallelize algorithms
Interconnect delay/power trends

- Interconnect delay scales slower than gate delay
  - Global interconnects could become slower

<table>
<thead>
<tr>
<th>Technology (um)</th>
<th>0.25</th>
<th>0.18</th>
<th>0.15</th>
<th>0.13</th>
<th>0.10</th>
<th>0.07</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intrinsic gate delay (ns)</td>
<td>0.071</td>
<td>0.051</td>
<td>0.049</td>
<td>0.045</td>
<td>0.039</td>
<td>0.022</td>
</tr>
<tr>
<td>1mm (ns)</td>
<td>0.059</td>
<td>0.049</td>
<td>0.051</td>
<td>0.044</td>
<td>0.052</td>
<td>0.042</td>
</tr>
<tr>
<td>2cm no-opt (ns)</td>
<td>2.589</td>
<td>2.48</td>
<td>2.65</td>
<td>2.62</td>
<td>3.73</td>
<td>4.67</td>
</tr>
<tr>
<td>2cm best-opt (ns)</td>
<td>0.895</td>
<td>0.793</td>
<td>0.77</td>
<td>0.7</td>
<td>0.77</td>
<td>0.672</td>
</tr>
</tbody>
</table>

Source: ITRS Roadmap

Delay for Metal 1 and Global Wiring versus Feature Size
Traditional Design Flow: Separation of Logic and Physical Design

- Earlier steps use approximate prediction models for interconnect (e.g., wire load models in logic synthesis)
The Timing Closure Problem

- Sometimes, timing appears to be OK during earlier stages, but fails during later stages
  - Need to repeat / iterate one or more stages to fix timing problems

Example:

White lines indicate timing violations

Source: Synopsys
Addressing Timing Closure

- Tighter integration of logic synthesis and physical design
  - Placement during technology mapping
  - Constant-delay synthesis
Nanometer Design Challenges: Variation in Process Parameters (Source: Kaushik Roy)

Inter and Intra-die Variations

Device 1

Device 2

Channel length

Normalized Frequency

Normalized Leakage (Isb)

Delay and Leakage Spread

Random dopant fluctuation

Device parameters are no longer deterministic
Impact of Process Variations

- Need to consider statistical models of delay, power!

SoC architecture

Micro-architecture

Gates

Devices & Interconnects

Applications

Statistical performance and power behavior of components

Variations in system behavior

Transistor & Interconnect characteristics

Random Dopant Fluctuation

Line Edge/Width Roughness

Lithographic effects

Frame Rate

Battery life variation

Time

Variations in system behavior

Statistical performance and power behavior of components
Dealing with Variations: Statistical Timing Analysis

• Traditional approach: Corners
  – Use best-case, worst-case, and typical-case values

• Problem: As spread of distribution increases, this leads to highly conservative estimates, incorrect critical paths, and over-design
Statistical Timing Analysis and Design

- Model delays of gates as probability distributions
- Statistical Timing Analysis: Compute delay distribution for entire circuit
- Statistical Optimization: Improve design considering overall delay distribution rather than worst-case delay
Parallelizing EDA

• EDA tools cope with increase in complexity by leveraging better algorithms and faster computers
  – Performance via parallelism is the new paradigm
• Focus on parallelizing the core algorithms in EDA
  – Graph algorithms (sis)
  – Backtrack / B&B (SAT, ATPG, ESPRESSO)
  – Dynamic Programming
  – Linear Algebra (Placement & Routing, Circuit Simulation)

Optimization Agents acting on a repository: A model for parallel EDA

Summary

• Digital Logic Synthesis is a mature area
  – Strong foundation in algorithms and optimization techniques
• Evolving nature of VLSI (scaling driven trends) keeps it a vibrant and active research area